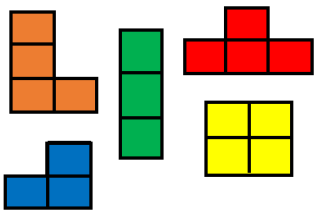
به نام خدا

پروژه پایانی برنامه‌نویسی پیشرفته



« بازی تتریس (خانه سازی) »

فهرست مطالب

۱– مقـــــــــدمه

- پروژه‌ی جاری باید به صورت **انفرادی** و یا حداکثر **دو نفره** پیاده‌سازی شود.
- برای پیاده‌سازی از نرم‌افزار **IntelliJ Idea** استفاده شود.
- ارائه‌ی پروژه در مکان و زمان مشخص شده در کلسروم خواهد بود.
- مهلت تقریبی پروژه، یک هفته تا یک ماه قبل از مهلت تایید نهایی نمرات است.
- نمره‌ی هر دانشجو، بسته به تسلط روی کد برنامه‌اش و کامل بودن برنامه‌اش محاسبه خواهد شد.
- ارائه با کامپیوتر شخصی دانشجو انجام می‌شود.

۲– صورت مسئله

پروژه‌ی پایانی این درس، بازی تتریس است که دارای قابلیت‌های کلی زیر است. این بازی به صورت تمام گرافیکی و تمام شیءگرا نوشته می‌شود. نرم افزار موردنظر یک Java Application (یک برنامه عادی) است نه یک Applet یا ... .

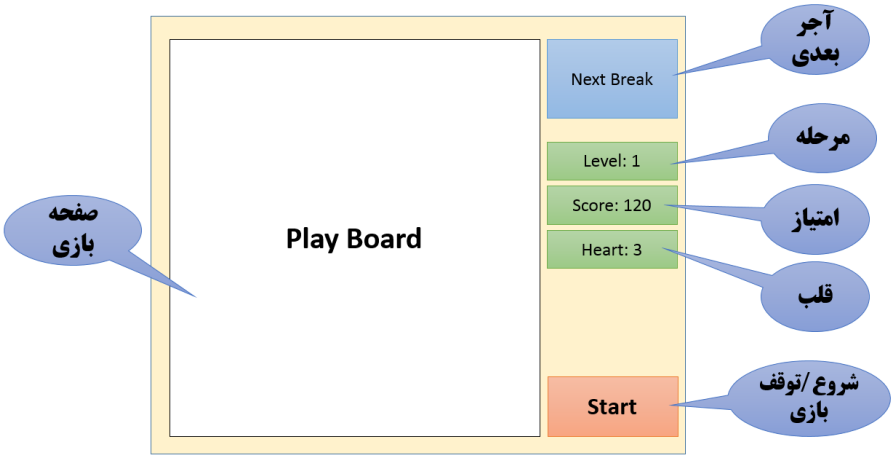
- منوی اصلی بازی
- صفحه اصلی بازی
- دارای امتیاز
- دارای قلب
- دارای چند مرحله
- قابلیت ثبت امتیاز برای هر بازیکن

۳– رابط کاربری نرم‌افزار

نرم‌افزار پس از اجرا باید دارای شکل تقریبی نشان داده شده در زیر باشد.

۳-۱- New Game

در صورتی که کاربر بر روی این گزینه کلیک کند، صفحه اصلی بازی ظاهر می‌شود. با کلیک کردن بر روی دکمه Start در صفحه اصلی بازی، بازی آغاز می‌شود. این صفحه به صورت تقریبی، باید شبیه به شکل زیر باشد.



۳-۲- Scores Table

در این صفحه، ۱۰ بازیکنی که بیشترین امتیاز را کسب کرده‌اند، به همراه نامشان نمایش داده می‌شوند. شکل تقریبی این صفحه باید به صورت زیر باشد.

Scores Table

Rank	Name	Score
1	Player1	1340
2	Player2	1205
3	Mina	1100
4	Mohammad	950
5	Sina	800
6	Elham	300
7	Ali	130
8		
9		
10		

۳-۳- Settings

در این بخش، تنظیمات بازی که موارد زیر را شامل می‌شود، قابل تغییرند. شکل تقریبی این پنجره نیز در تصویر نمایش داده شده است.

- **امتیاز کسب شده برای هر ردیف پر شده**: یک مقدار بین ۱۰ تا ۱۰۰۰ (مقدار پیشفرض ۱۰)
- **تنظیم رنگ پس زمینه بازی**: یک رنگ (مقدار پیشفرض: دلخواه)
- **مرحله‌ی آغازین بازی**: یک عدد بین ۱ تا ۱۰ (مقدار پیشفرض: ۱)
- **پاک کردن امتیازات ذخیره شده**: یک دکمه

Settings

Scores per row: 10

Background color:

Start level: 1

Clear Scores

۴– ذخیره‌سازی اطلاعات

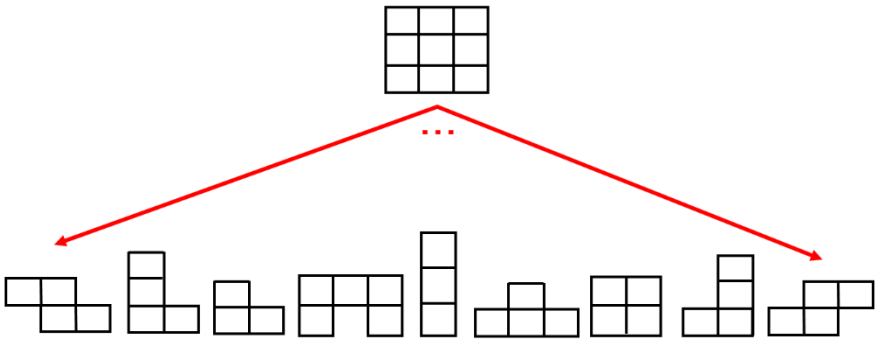
تمامی اطلاعات مربوط به بازی از قبیل تنظیمات و امتیازهای کسب شده، در فایل ذخیره می‌شوند.

- برای اطلاعات مربوط به تنظیمات یک فایل مجزا به نام settings.txt درنظر بگیرید.
- برای اطلاعات مربوط به امتیازات یک فایل مجزا به نام scores.txt درنظر بگیرید.

۵– جزئیات پیاده‌سازی (منطق)

۵-۱- کلاس آجر (Brick)

در این بازی، ۹ نوع آجر وجود دارد که در شکل نشان داده شده‌اند. همه آجرها از کلاس آجر عمومی (Brick) به ارث می‌برند.



آجر پدر دارای فیلدها و توابع زیر است:

- فیلد x و y: مکان آجر در صفحه بازی را نگه می‌دارد
- دارای دو فیلد ثابت طول و عرض (با مقدار ۳)
- دارای یک فیلد رنگ (برای هر نوع آجر، یک رنگ منحصربفرد درنظر بگیرید)
- یک آرایه ۳ در ۳ از نوع Boolean: مشخص می‌کند کدام خانه‌ها پر هستند.
- سازنده با یک پارامتر: صفحه بازی را دریافت می‌کند
- سازنده با پارامتر صفحه بازی و x و y
- توابع setter و getter موردنیاز
- تابع move: این تابع یک متغیر شمارشی از نوع Direction دریافت کرده و به چپ، راست یا پایین (یک واحد) حرکت می‌کند.
- تابع paint: این تابع خود (آجر) را بر روی صفحه بازی رسم می‌کند.
- تابع rotate: در جهت ساعتگرد، ۹۰ درجه می‌چرخد.
- تابع freeze: خانه‌های مربوط به خود را در آرایه صفحه بازی برابر با true قرار می‌دهد (کاربرد: زمانی که آجر به انتهای صفحه بازی می‌رسد)

نکات:

- هر آجر تابع rotate را برای خود دوباره‌نویسی (Override) می‌کند.
- نوع شمارشی Direction دارای سه ثابت LEFT و RIGHT و DOWN است.
- توابع move و rotate در صورت موفقیت در انجام حرکت/چرخش، مقدار true را برمی‌گردانند.

۵-۲- کلاس صفحه بازی (GameBoard)

صفحه بازی یک کلاس مجزا است که دارای فیلدها و توابع زیر است. اگر فیلد یا تابع دیگری نیاز داشتید، می‌توانید اضافه کنید؛ البته در صورتی که خلاف اصول طراحی توضیح داده شده نباشد.

- یک آرایه دو بعدی به اندازه‌ی ابعاد صفحه بازی از نوع Boolean که تعیین می‌کند کدام خانه پر است و کدام خانه خالی.
- فیلد state: از نوع شمارشی State که دارای دو ثابت START و STOP است.
- فیلد رنگ پس زمینه: از نوع رنگ
- فیلد مرحله: یک مقدار بین ۱ تا ۱۰
- دو فیلد ثابت تعداد سطرها و ستون‌ها (با مقادیر ۲۰ و ۱۵)
- فیلد ثابت اندازه سلول (هر خانه برابر با ۲۰ پیکسل است: هنگام رسم آجرها کاربرد دارد)
- سازنده پیشفرض ندارد.
- سازنده با دو پارامتر کلاس اصلی بازی و مرحله بازی: یک شیء از کلاس اصلی بازی (در بخش بعد توضیح داده شده است) دریافت می‌کند تا در مواقع نیاز بتواند با آن ارتباط برقرار کند؛ و یک عدد که مرحله بازی را مشخص می‌کند.
- تابع start: بازی آغاز می‌شود. اگر بازی قبلا در نیمه‌های خود متوقف شده است، ادامه می‌یابد (آجری که در نیمه راه بوده است از بین رفته و آجری جدید تولید می‌شود). در غیر اینصورت از ابتدا شروع به کار می‌کند (یک آجر تصادفی تولید شده و شروع به حرکت می‌کند).
- تابع stop: بازی متوقف می‌شود.
- تابع moveDown(): آجر را با سرعت به پایین حرکت می‌دهد.
- تابع move(Direction): آجر را به سمت چپ یا راست حرکت می‌دهد.
- تابع isFilled(x, y): یک مختصات دریافت کرده و در صورتی که آن خانه پر باشد، true برمی‌گرداند
- تابع fill(x, y): یک مختصات دریافت کرده و آن خانه را پر می‌کند
- تابع paint: خود را بر روی کنترل موردنظر رسم می‌کند

نکات:

- سرعت حرکت آجرها بسته به مرحله بازی تنظیم می‌شود.

۵-۳- کلاس بازی (Tetris): کلاس اصلی بازی

کلاس اصلی بازی شامل فیلدها و توابع زیر است:

- یک شیء از کلاس صفحه اصلی بازی
- یک شیء از کلاس تنظیمات
- یک شیء از کلاس امتیازها
- فیلد امتیاز
- فیلد قلب
- فیلد مرحله

- سازنده پیشفرض
- توابع setter و getter موردنیاز
- تابع load: تابع load مربوط به شیء کلاس تنظیمات (SettingsManager) و امتیازها (ScoresManager) را فراخوانی می‌کند.
- تابع save: تابع save مربوط به شیء کلاس تنظیمات و امتیازها را فراخوانی می‌کند.
- تابع moveDown: تابع moveDown مربوط به صفحه بازی را فراخوانی می‌کند.
- تابع move: تابع move مربوط به صفحه بازی را فراخوانی می‌کند.
- تابع start: تابع start مربوط به صفحه بازی را فراخوانی می‌کند.
- تابع stop: تابع stop مربوط به صفحه بازی را فراخوانی می‌کند.

نکات بسیار مهم:

- گرافیک بازی کاملا مستقل از منطق بازی است. همه کلاس‌هایی که در این سند توضیح داده شد، بخش منطق بازی را پوشش می‌دهد.
- در پروژه‌ی اصلی، تنها یک شیء از این کلاس ایجاد می‌شود و همه امور توسط این کلاس مدیریت می‌شود. این کلاس خود از سایر کلاس‌ها بهره می‌برد.

۵-۴- کلاس تنظیمات (SettingsManager)

کلاس تنظیمات دارای فیلدهای گفته شده در بخش تنظیمات است. و دارای توابع setter و getter برای همه فیلدهای خود است. علاوه‌بر این موارد، توابع زیر نیز لازم است:

- تابع load: اطلاعات ذخیره شده در فایل را خوانده و در فیلدها قرار می‌دهد
- تابع save: اطلاعات فیلدها را در فایل ذخیره می‌کند.

۵-۵- کلاس امتیازها (ScoresManager)

این کلاس دارای آرایه‌ای از کلاس Player است و دارای توابع زیر است:

- اضافه کردن Player: در صورتی که امتیاز بازیکن در بین ده امتیاز برتر قرار داشته باشد، در مکان مناسب اضافه می‌شود.
- پاک کردن همه: فهرست بازیکن‌ها پاک می‌شود.
- تابع load: اطلاعات ذخیره شده در فایل را خوانده و در فیلدها قرار می‌دهد.
- تابع save: اطلاعات فیلدها را در فایل ذخیره می‌کند.

۵-۶- کلاس Player

این کلاس دارای دو فیلد و دو سازنده زیر است و برای هر یک از فیلدها دارای توابع setter و getter می‌باشد.

- امتیاز
- نام
- سازنده پیشفرض
- سازنده با دو پارامتر نام و امتیاز

۶– جزئیات پیاده‌سازی (گرافیک)

برای طراحی گرافیکی بهینه، باید یک نگاشت مناسب بین هر یک از بخش‌های منطقی و گرافیکی یافت. همه پنجره‌های اصلی از نوع JFrame (Scene در JavaFX) هستند و طراحی آنها ساده است. در ادامه به صفحه اصلی بازی می‌پردازیم که از پیچیدگی نسبی برخوردار است. همانطور که در شکل زیر مشاهده می‌کنید، پنجره اصلی بازی از بخش‌های زیر تشکیل شده است. در این شکل دو JPanel سفارشی دیده می‌شود. توضیحات داده شده مبتنی بر Swing هستند. شما با توجه به شناختتان از JavaFX، معادل مناسب در JavaFX را به کار بگیرید.

JPanel سمت چپ مهمترین نقش را دارد و صفحه بازی بر روی آن رسم خواهد شد. JPanel سمت راست (و بالا) برای نمایش آجر بعدی است. بین JPanel سمت چپ و کلاس صفحه بازی (GameBoard) باید یک رابطه مستقیم برقرار باشد. هر دوی این کلاس ها باید یک ارجاع از هم در اختیار داشته باشند. هرگاه که کلاس GameBoard نیاز به رسم خود در صفحه داشت، باید به JPanel درخواست دهد تا او این کار را انجام دهد. خود JPanel با دوباره نویسی تابع paintComponent، کلاس GameBoard را رسم می‌کند.

اندازه JPanel سمت چپ به صورت دقیق قابل تعیین است. تعداد سطر و ستون‌های صفحه بازی مشخص است و همچنین تعداد پیکسل‌هایی که هر سلول اشغال می‌کند. در نتیجه عرض این کنترل برابر خواهد بود با "تعداد ستون ها * اندازه هر سلول" و همچنین تعداد سطرها برابر خواهد بود با "تعداد سطرها * اندازه هر سلول"

با فشردن کلیدهای چپ و راست، آجر فعلی به چپ و راست حرکت می‌کند. با فشردن کلید بالا، آجر فعلی در جهت ساعتگرد، ۹۰ درجه می‌چرخد. با فشردن کلید Enter، آجر فعلی به سمت پایین سقوط می‌کند. (از گره زدن کلید استفاده کنید).

۷– تحویل پروژه

- تحویل پروژه به صورت انفرادی صورت می‌گیرد.
- در صورتی که پروژه به صورت دو نفره نوشته شده باشد، هر دو نفر باید تسلط کامل بر همه‌ی بخش‌های آن داشته باشند. در صورتی که هر یک از دو نفر، اشکالی در توضیح هر یک از بخش‌های برنامه داشته باشد، مقداری از نمره‌ی کل گروه کسر خواهد شد.
- در پیاده‌سازی بخش‌هایی از پروژه که نحوه پیاده‌سازی آن‌ها صریحا بیان شده، **تنها** مجاز به استفاده از دستورات و مطالبی هستید که در کلاس آموزش داده شده است.
- برای بخش‌هایی که صحبتی در رابطه با نحوه پیاده‌سازی آن‌ها نشده، می‌توانید نوآوری به خرج داده و از ایده‌های خود استفاده کنید.
- در صورت یافتن مشابهت در پروژه‌ی دو گروه، نمره‌ی پروژه به یکی از گروه‌ها (به صورت تصادفی) داده شده و نمره **صفر** به گروه دیگر تعلق خواهد گرفت.

موفق باشید، بیگلری